(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平11－194899

(43)公開日　平成11年(1999)７月21日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	3/06	５４０	Ｃ０６Ｆ	3/06	５４０
		３０５			３０５Ｃ
Ｇ１１Ｂ	20/10		Ｇ１１Ｂ	20/10	Ａ
	20/18	５７０		20/18	５７０Ｚ

審査請求　未請求　請求項の数15　ＦＤ　（全 16 頁）

(21)出願番号　特願平9－366782

(22)出願日　平成９年(1997)12月26日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　冨田　治男
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

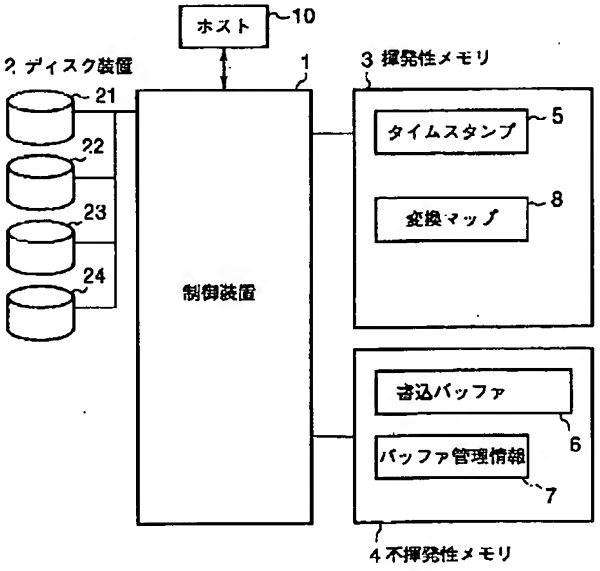
(74)代理人　弁理士　鈴江　武彦　（外６名）

(54)【発明の名称】　ディスク記憶システム及び同システムに適用するデータ更新方法

(57)【要約】

【課題】高速の書込み方法を採用したＲＡＩＤ方式のディスク記憶システムに適用して、特にデータ更新処理に関する性能の向上を図ることにある。

【解決手段】ＲＡＩＤ構成のディスク記憶システムであって、ストライプに相当する記憶容量を有する書込みバッファ６と、バッファ管理テーブル７と、制御装置１とを有し、制御装置１は書込みバッファ６に必要に応じてデータ長を可変した論理ブロックを蓄積し、データ更新処理時に書込みバッファ６に蓄積した論理ブロックがＮ＊Ｋ－１個に達するまでその論理ブロックの更新を遅延させ、Ｎ＊Ｋ－１個の論理ブロックに論理アドレス・タグ・ブロックを加えたＮ＊Ｋ個の論理ブロックを、旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込む。



【特許請求の範囲】

【請求項1】　N台のディスクドライブから構成されるディスク記憶システムであって、

N＊K（Kはブロック数を意味する整数）個の論理ブロックに相当する記憶容量を有する書込みバッファ手段と、

前記書込みバッファを前記各ディスクドライブとの間で転送される読出しデータ及び書込みデータのキャッシュメモリとして管理し、前記書込みバッファに必要に応じてデータ長を可変した前記論理ブロックを蓄積するための管理手段と、

データ更新処理時に前記書込みバッファに蓄積した論理ブロックがN＊K－1個に達するまでその論理ブロックの更新を遅延させ、当該N＊K個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むための書込み制御手段とを具備したことを特徴とするディスク記憶システム。

【請求項2】　前記管理手段は前記書込みバッファ手段と共にメモリ上に構成されたバッファ管理テーブルを有し、書込みデータをデータ長に応じて最適なブロック単位に分割し、前記書込みバッファ手段にログ形式で順番に詰めて格納し、ホストシステムからの書込みデータの論理アドレスを前記書込みバッファ手段の格納領域に対応する前記バッファ管理テーブルのエントリに格納するように構成されたことを特徴とする請求項1記載のディスク記憶システム。

【請求項3】　前記書込み制御手段は、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成し、N＊K－1個の論理ブロックに前記論理アドレス・タグ・ブロックを加えたN＊K個の論理ブロックを前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域に書込みむように構成されたことを特徴とする請求項1記載のディスク記憶システム。

【請求項4】　N台のディスクドライブから構成されるディスク記憶システムに適用するデータ更新方法であって、

N＊K（Kはブロック数を意味する整数）個の論理ブロックに相当する記憶容量を有する書込みバッファ手段に前記論理ブロックのデータ長を必要に応じて可変長で可変した前記論理ブロックを蓄積するステップと、

データ更新時に前記書込みバッファ手段に蓄積した前記論理ブロックがN＊K－1個に達するまで、当該論理ブロックの更新を遅延させるステップと、

当該N＊K個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むステップとからなる処理を実行することを特徴とするデータ更新方法。

【請求項5】　N台のディスクドライブから構成されるディスク記憶システムに適用するデータ更新方法であって、

N＊K（Kはブロック数を意味する整数）個の論理ブロックに相当する記憶容量を有する書込みバッファ手段に前記論理ブロックのデータ長を必要に応じて可変長で可変した前記論理ブロックを蓄積するステップと、

データ更新時に前記書込みバッファ手段に蓄積した前記論理ブロックがN＊K－1個に達するまで、当該論理ブロックの更新を遅延させるステップと、

前記書込みバッファ手段に蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成するステップと、

当該N＊K－1個の論理ブロックに前記論理アドレス・タグ・ブロックを加えたN＊K個の論理ブロックを生成するステップと、

前記論理アドレス・タグ・ブロックを加えたN＊K個の論理ブロックを前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むステップとからなる処理を実行することを特徴とするデータ更新方法。

【請求項6】　N台のディスクドライブから構成されるディスク記憶システムに適用するデータ更新方法であって、

N＊K（Kはブロック数を意味する整数）個の論理ブロックに相当する記憶容量を有する書込みバッファ手段に前記論理ブロックのデータ長を必要に応じて可変長で可変した前記論理ブロックを蓄積するステップと、

データ更新時に前記書込みバッファ手段に蓄積した前記論理ブロックがN＊K－1個に達するまで、当該論理ブロックの更新を遅延させるステップと、

前記書込みバッファ手段に蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成するステップと、

前記N＊K－1個の論理ブロックに前記論理アドレス・タグ・ブロックを加えたN＊K個の論理ブロックからK個のパリティブロックを生成するステップと、

当該パリティブロックを加えたN＊K個の論理ブロックを生成するステップと、

前記パリティブロックを加えたN＊K個の論理ブロックを前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むステップとからなる処理を実行することを特徴とするデータ更新方法。

【請求項7】　前記書込みバッファ手段に対する書込み操作を行なう場合には、論理アドレスに対応してディス

クドライブ上の連続したセクタの位置に対応するように論理ブロックを再配置するステップを含むことを特徴とする請求項４、請求項５、請求項６のいずれか記載のデータ更新方法。

【請求項８】　前記Ｎ＊Ｋ個の論理ブロックをストライプと定義した場合に、前記ディスク装置上に連続した論理ブロックを連続して書込めるような空領域を作るため、前記複数のストライプを読出して最新の論理ブロックだけを前記書込みバッファ手段に移して、対応する論理アドレス・タグ・ブロック内の論理アドレスから新しい論理アドレス・タグ・ブロックを生成するステップと、

前記書込みバッファ手段の有効データと生成された論理アドレス・タグ・ブロックとから構成されるストライプを、読出したストライプを保持していた領域とは別の空領域に依存するストライプの物理アドレスに順次書込むステップとを含む請求項４、請求項５、請求項６のいずれか記載のデータ更新方法。

【請求項９】　前記空領域をつくるために、データの読出し要求や書込み要求とは独立した動作時に、読出した前記ストライプを保持していた領域とは別の空領域のデータの配置を変更するステップを含むことを特徴とする請求項４、請求項５、請求項６のいずれか記載のデータ更新方法。

【請求項１０】　ホストシステムからの論理アドレスと前記ディスクドライブの物理アドレスとを変換するめたの変換マップを設けて、アドレス変換時に前記変換マップの全てのエントリを検索するステップを含むことを特徴とする請求項４、請求項５、請求項６のいずれか記載のデータ更新方法。

【請求項１１】　前記アドレス変換時に前記変換マップから論理アドレスをハッシュキーにした線形検索を実行するステップを含むことを特徴とする請求項１０記載のデータ更新方法。

【請求項１２】　前記論理アドレスから前記物理アドレスへのアドレス変換時に、前記変換マップを木構造で管理し、二分木検索を実行するステップを含むことを特徴とする請求項１０記載のデータ更新方法。

【請求項１３】　前記各論理アドレスに対するストライプ番号、ストライプ内のブロック番号、有効データのタイムスタンプから構成される前記変換マップを生成し、前記論理アドレス・タグ・ブロックに基づいて変換マップの一部を前記ディスクドライブスク内の空領域に格納するステップを含むことを特徴とする請求項１０記載のデータ更新方法。

【請求項１４】　前記変換マップの一部を前記ディスクドライブに格納する場合に、最も最後にアクセスされたデータを含む論理ブロックに対応する領域に応じて順次格納するステップを含むことを特徴とする請求項１３記載のデータ更新方法。

【請求項１５】　前記書込み制御手段は、前記ディスクドライブ上のセクタ位置に対応して、前記ディスクドライブ上の更新対象の旧データを保持している領域を選択して記憶領域に書込むことを特徴とする請求項４、請求項５、請求項６のいずれか記載のデータ更新方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、特にＲＡＩＤ（Redundant Array of Inexpensive Disk ）と呼ぶディスクアレイ装置から構成されるディスク記憶システムに適用し、具体的にはデータ更新機能を改善したディスク記憶システムに関する。

【０００２】

【従来の技術】従来、複数のディスクドライブから構成されるディスクアレイ装置（ＲＡＩＤと呼ぶ）が周知である。ＲＡＩＤは、例えばネットワーク・サーバに接続されて、大容量のデータを記憶できるディスク記憶システムである。このようなＲＡＩＤに関して、特に高速のデータ書込み方法として、データ更新処理時に更新データを旧データの記憶領域に書換えることなく、更新データを一括して書込む方法が提案されている。具体的には、更新データを用意した書込みバッファに蓄積し、この蓄積した更新データをディスクドライブ内に予め用意した空領域に一括して書込む方法である（特開平６－２１４７２０号公報、特開平６－２６６５１０号公報、特願平９－２１４６４５号公報を参照）。

【０００３】図２２を参照して、従来の書込み方法（データ更新方法）について簡単に説明する。図２２において、Ｌ１～Ｌ９９は論理ブロックを意味し、Ｐ１～Ｐ５６はディスク装置（ディスクドライブ）内の物理ブロックを意味する。ここでは、論理ブロックＬ６，Ｌ４，Ｌ２，Ｌ１２，Ｌ７，Ｌ１１のそれぞれを更新する場合を想定する。これらの論理ブロックの各旧データは、ディスク装置内の物理ブロックＰ６，Ｐ４，Ｐ２，Ｐ１２，Ｐ７，Ｐ１１に存在する。従って、通常の書込み方法では、これらの物理ブロックＰ６，Ｐ４，Ｐ２，Ｐ１２，Ｐ７，Ｐ１１の内容を更新することになる。しかし、高速の書込み方法では、物理ブロックＰ６，Ｐ４，Ｐ２，Ｐ１２，Ｐ７，Ｐ１１の各データはそのまま維持して、新しい論理ブロックＬ６，Ｌ４，Ｌ２，Ｌ１２，Ｌ７，Ｌ１１の各データを予め用意した別の空領域である物理ブロックＰ５１，Ｐ５２，Ｐ５３，Ｐ５４，Ｐ５５，Ｐ５６に一括して書込む。これにより、通常の書込み方法では６回の書込み操作が必要であるが、高速の書込み方法では例えば２物理ブロックを一括して書込む場合には３回の書込み操作に減少させることが可能であり、書込み性能を向上させることができる。

【０００４】ここで、図２２に示す具体例では、便宜的に１ディスクに２物理ブロックを一括して書込む場合を示しているが、実際には数十物理ブロックが一括して書

込まれる。また、レベル4のＲＡＩＤ方式（ＲＡＩＤ４方式）およびレベル5のＲＡＩＤ方式（ＲＡＩＤ５方式）では、１個のストライプ（Ｎ＊Ｋ個の論理ブロックであり、Ｎはドライブ数、Ｋはブロック数）を一括して書換えることが可能であるため、パリティ維持のためのディスク読出し動作も不要であり、書込み時のオーバヘッドを減少させることも可能である。なお、ＲＡＩＤ４方式は、データの配置単位をビットやバイトのような小さな単位ではなく、セクタやブロックのような大きな単位とし、小容量のデータ読出し要求に対してディスクを独立動作可とする方式である。また、ＲＡＩＤ５方式は、冗長データ（パリティデータ）を専用のパリティディスクに格納するのではなく、各データディスクに巡回的に配置する方式である。

【０００５】一方、論理ブロックＬ６、Ｌ５、Ｌ２、Ｌ１２、Ｌ７、Ｌ１１の各最新データはディスク装置内の物理ブロックＰ５１～Ｐ５６に存在するので、間接マップの内容を正しいディスク位置を指示するように書き換える。また、論理ブロックのデータを読出すときには、この間接マップを参照して、最新のディスク位置を求めてから読出すので、旧データを読出すような不都合は発生しない。

【０００６】

【発明が解決しようとする課題】前記のような従来の高速の書込み方法では、図２２に示すように、不揮発性メモリ上に書込みバッファを用意し、この書込みバッファを介してディスク装置内の空領域をブロック単位に分割し、空ブロックから順番に詰めて格納している。このため、読出すデータが連続である場合にもブロック単位に分割してから、ブロック毎のアドレスに変換する必要がある。従って、ホストシステムから論理ブロックのデータ長を越える読込み要求があった場合に、当該要求データは複数の論理ブロックに分割されるという問題点がある。

【０００７】また、特にデータ更新処理においては、毎回更新すべきデータがバッファ管理テーブルに対応するエントリに存在するかどうかを変換マップを使用して検索する必要があるため、この検索に要する分だけ処理時間が増大する問題がある。さらに、変換マップはディスク装置の記憶容量に依存して、システムのメインメモリ上に作成する必要があるため、当該メインメモリの使用容量を制限する要因になっている。

【０００８】そこで、本発明の目的は、特にＲＡＩＤ方式のディスク記憶システムにおいて、高速の書込み方法を適用した場合のデータ更新処理に関する性能の向上を図ることにある。

【０００９】

【課題を解決するための手段】本発明は、特にＲＡＩＤ方式のディスク記憶システムにおいて、連続したデータの読出し要求に応じた読出し処理時に、原理的に論理ブロックを分割することなく読出し処理を実行する方式である。また、データ更新処理時にバッファ管理テーブルに対する変換マップのエントリを検索するための検索処理の効率化を実現する方式である。

【００１０】具体的には、本発明は、Ｎ台のディスクドライブから構成されるディスク記憶システムであって、Ｎ＊Ｋ（Ｋはブロック数を意味する整数）個の論理ブロックに相当する記憶容量を有する書込みバッファ手段と、前記書込みバッファを前記各ディスクドライブとの間で転送される読出しデータ及び書込みデータのキャッシュメモリとして管理し、前記書込みバッファに必要に応じてデータ長を可変した前記論理ブロックを蓄積するキャッシュ管理手段と、データ更新処理時に前記書込みバッファに蓄積した論理ブロックがＮ＊Ｋ－１個に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成し、Ｎ＊Ｋ－１個の論理ブロックに前記論理アドレス・タグ・ブロックを加えたＮ＊Ｋ個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域（セクタ単位）を選択して、連続した書込み操作により順次書込むためのデータ更新手段とを有するシステムである。

【００１１】また、本発明の別の観点として、前記データ更新手段は、データ更新処理時に前記書込みバッファに蓄積した論理ブロックがＮ＊Ｋ－１個に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成し、Ｎ＊Ｋ－１個の論理ブロックに前記論理アドレス・タグ・ブロックを加えたＮ＊Ｋ個の論理ブロックからＫ個のパリティブロックを生成し、この論理ブロックにパリティブロックを加えたＮ＊Ｋ個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域（セクタ単位）を選択して、連続した書込み操作により順次書込む機能を有するように構成されたディスク記憶システムである。

【００１２】このような構成により、読出すデータが連続である場合には、不揮発姓メモリ上に設ける書込みバッファの空領域を論理ブロック単位に分割する必要性を無くすことが可能となる。

【００１３】

【発明の実施の形態】以下図面を参照して本発明の実施の形態を説明する。

【００１４】図１及び図２３は本実施形態のＲＡＩＤ方式のディスク記憶システムの概念的構成を示すブロック図である。なお、本発明に関する記述の全体を通じて、より詳細に説明するために図１に示す実施形態を中心と

して説明するが、当該技術分野に属する熟練者であれば、図１及び図２３から派生するシステム構成であって、本発明を実施することが可能であればよい。例えば、図１において、ホストシステム１０と制御装置１とのインターフェースは、ＳＣＳＩインターフェースであっても良いし、ＰＣＩインターフェースであってもよい。なお、図２３のシステムは、図１に示すバッファ管理テーブル７に相当するバッファ管理情報２３－７を揮発性メモリ３である主記憶メモリ２３－３に設けた場合であり、ほかの構成要素は図１に示すシステムと同様である。

（システム構成）本システムは、図１に示すように、制御装置（ＣＰＵ）１、ディスク装置２（ディスクドライブ２１，２２，２３，２４）、揮発性メモリ３、および不揮発性メモリ４から構成される。揮発性メモリ３には、書込の時間的順序を維持するためのタイムスタンプ５、論理アドレスからディスク装置２へアクセスするための物理アドレスへの変換を行なうための変換マップ８が設けられている。また、不揮発性メモリ４には、ディスク装置２に書込むデータをログ構造化して保持するための書込みバッファ６として使用する領域、およびバッファ管理情報（バッファ管理テーブル）７を格納する領域が設けられている。バッファ管理情報７は、書込みバッファ６内に保持されている書込みデータの論理アドレスを管理するためのテーブル情報である。制御装置１は、タイムスタンプ５、書込みバッファ６、およびバッファ管理情報７を管理することにより、ディスク装置２に対するデータの入出力を制御する。

（書込みバッファ６とバッファ管理情報７との関係）図２から図５は、不揮発性メモリ４に割り付けられた書込みバッファ６とバッファ管理情報７との関係を示す図である。

【００１５】本実施形態のシステムでは、制御装置１は、外部のホストシステム１０から書込み要求された書込みデータを、ディスク装置２に対して即書込み処理せずに、当該書込みデータをデータ長に応じて最適なブロック単位に分割し、書込みバッファ６にログ形式で順番に詰めて格納する。このとき、ホストシステム１０からの書込みデータの論理アドレスを、書込みバッファ６の格納領域に対応するバッファ管理テーブル７のエントリに格納する。

【００１６】バッファ管理テーブル７のエントリには、エントリの状態を示すフラグ（Ｆ，Ｕ，Ｃ）が設定される。フラグＦは、エントリにデータが割り当てられた事を示す。フラグＵは、エントリが使用されていない状態を示す。フラグＣは次のエントリが連続したデータにより割り当てられている状態を示す。制御装置１は、バッファ管理テーブル７を参照することにより、ホストシステム１０から受取った書込みデータを格納すべき次の書込みバッファ６の格納位置を決定できる。図２に示す具体例では、書込みバッファ６のバッファ領域（格納位置）Ｂ０からＢ７まで書込みデータが格納されている。ＬＡ９９、ＬＡ１００、Ｌ３５、Ｌ６７８…、ＬＡ５４１は、格納されている書込みデータの論理アドレスを意味する。

【００１７】ここで、本実施形態では、書込みバッファ６からディスク装置２への書込み処理はストライプ単位であり、読込み処理は可変長の論理ブロック単位であると想定する。ストライプ単位の書込み処理とは、Ｎ台のディスクドライブにおいて、Ｎ＊Ｋ（Ｋブロック数を示す整数）個の論理ブロックのデータの一括書込み動作の実行である。

（本実施形態の動作の説明）まず、図２を参照して、書込みバッファ６に格納された論理ブロックのデータの前の論理アドレスに位置する論理ブロックに、ホストシステム１０から書込み要求があった場合の動作を説明する。本実施形態では、説明を簡略にするために、書込みバッファを書込みバッファと読込みバッファとの共有バッファとして想定しているが、これらのバッファはそれぞれ独立であっても問題ない。この場合、当然ながら読込みバッファは揮発性メモリまたは不揮発性メモリのいずれにあってもよい。

【００１８】制御装置１は、ホストシステム１０から書込み要求された書込みデータに対応する論理アドレスＬＡ３４の場合において、バッファ管理テーブル７を参照して、次に書込み可能な書込みバッファ６のバッファ領域Ｂ８（フラグＵにより現時点では未格納領域）を認識する。このとき、書込み用の論理アドレスとして連続したバッファ領域を割り当てることが可能ように、書込みバッファ６上の格納位置（バッファ領域）を移動して、バッファ領域Ｂ２を格納位置として決定する。即ち、図３に示すように、制御装置１は、論理アドレスＬＡ３４のデータをバッファ領域Ｂ２に格納し、バッファ領域Ｂ８まで順次移動してデータを格納する。

【００１９】次に、書込みバッファ６に格納された論理ブロックのデータ更新処理において、２論理ブロックにまたがる書込み要求の場合の動作を説明する。

【００２０】ホストシステム１０からから書込み要求された書込みデータに対応する論理アドレスＬＡ１００の場合において、制御装置１は、図２に示すバッファ管理テーブル７を参照して、次の書込み可能なバッファ領域Ｂ８及び論理アドレスＬＡ１００に対応するバッファ領域Ｂ１を認識する。ここで、データ更新処理として、２論理ブロックにまたがるため、論理アドレスＬＡ１００のデータサイズが増えた分の論理アドレスＬＡ１０１に対応するバッファ領域を割り当てる必要がある。この場合には、遅延したストライプ毎の書込み処理時に、ストライプ上の物理ブロックとして連続に割り当てるようにする。このときのブロック移動は最小ですむようにすると効率が良い。このデータ更新処理では、図４に示すよ

うな状態となる。

【００２１】さらに、論理アドレスとして連続する２ブロックの読出し要求に対して、物理アドレスとして連続的に配置されていない場合に、物理アドレスとして連続化するための作を説明する。

【００２２】ホストシステム１０から要求された論理アドレスがＬＡ１１１、ＬＡ１１２の場合において、制御装置１は、図２に示すバッファ管理テーブル７を参照して、次の書込み可能なバッファ領域Ｂ８、及び論理アドレスとして連続したバッファ領域を割り当てることが可能であるバッファ領域Ｂ４、Ｂ５を決定する。ここでは、論理アドレスＬＡ１１１及びＬＡ１１２のそれぞれに対応するデータが、ディスク装置２上の別々のストライプに配置されている場合を想定するので、２回の読出し処理が必要となる。そこで、連続したブロックとしてまとめることにより、当該ストライプが書込まれた後では、読出し処理のためのディスク装置２への読込み要求は、同一ストライプ上の連続した物理ブロックへの要求を一度だけ実行すればよい。この場合の書込み処理では、図５に示すような状態となる。

【００２３】以上のように書込みバッファ６とバッファ管理テーブル７との関係について説明した。図２から図５に示す具体例では、書込みバッファ６に対する読込み処理や書込み処理時に連続した論理アドレスとして位置するデータに対して、物理的にも連続したアドレスになるようにデータの再配置が行なわれる。但し、コンピュータシステムのデータ転送の性能に基づいて、書込みバッファ６の全てのデータを論理アドレスの順番に並び替える方法でもよい。また、データを並び替える代わりに、書込みバッファ６からディスク装置２に書出すブロックの順を示したリストを備えることにより、データ移動を伴うことなく、リスト中の順番を変更する方法でもよい。

【００２４】また、ディスク装置２（ディスクドライブ２１～２４）は、それぞれブロックサイズの整数倍（Ｋ）であるストライプユニットと呼ぶ予め決められた単位（ディスク上の１トラック長に近いサイズが良い）で書込み処理を一括して実行する。このとき、ディスクドライブ２１～２４の物理的に同じ位置のストライプユニットは１つのストライプとして、同じタイミングで書込み処理が行われる。また、ホストシステム１０には、実際のディスクドライブ２１～２４を合わせて全記憶容量よりも少ない容量のディスク装置２として見られている。具体的には、ホストシステム１０が初期時にディスク装置２の記憶容量を問い合わせて来ると、制御装置１は実際の記憶容量より少ない記憶容量を応答する。従って、ホストシステム１０が論理的に書込み、読出し可能な記憶領域以外に、ディスク記憶システムとしては独自の記憶領域を確保することが可能となる。この記憶領域を、本実施形態では空領域と呼ぶことにする。

【００２５】さらに、タイムスタンプ５は、ホストシステム１０からの書込みデータが実際にディスク装置２に書込まれるときに付加される情報であり、ディスク装置２内でのデータ書込み順序を判定するのに使用される。書込みバッファ６のデータがディスク装置２に書込まれる毎に、タイムスタンプ５はインクリメントされることになる。

（システムの書込み動作）以下図１のシステムにおいて、本実施形態の書込み動作を主として図６を参照して説明する。

【００２６】制御装置１は、ホストシステム１０から書込みデータとその論理アドレスを受取ると、不揮発性メモリ４上の書込みバッファ６の空領域にブロック単位に分割して詰めて格納する。また、受取った論理アドレスはブロック毎のアドレスに変換して、バッファ管理テーブル７の対応するエントリに格納する。これらの処理は、受取った論理アドレスおよび書込みバッファ６に既に格納されているデータを参照することにより、書込みバッファ６の最適な格納位置に詰めて格納する。なお、既に書込みバッファ６に格納されているデータに対するデータ更新処理の場合には、書込みバッファ６の空領域に詰めて格納するのではなく、直接に書込みバッファ６内の旧データを更新する。

【００２７】ホストシステム１０からの書込みデータは、１ストライプ分に１ブロック少ない数（Ｎ＊Ｋ－１）だけ書込みバッファ６に蓄積された段階で、制御装置１はそれらのデータをディスク装置２に書込み処理する。このとき、最後の書込みブロックとして、バッファ管理テーブル７に格納された各ブロックの論理アドレスと揮発性メモリ３上のタイムスタンプ５から論理アドレス・タグ・ブロックを作成する。この論理アドレス・タグ・ブロック内のアドレスデータとデータブロックとの間には、１対１の関係があらかじめ設定されており、各データブロックの論理アドレスが分かるようになっている。

【００２８】この後に、制御装置１は、当該論理アドレス・タグ・ブロックを加えた１ストライプ分のデータを、一括してディスクドライブ２１－２４の空領域に同時に書き込む。前述したように、１ストライプ内の書込みブロックは、論理アドレスに対して連続した領域になっているため、最新のデータは同じストライプ上に連続した配置となるため、読込み性能が向上する。このような動作を図６に示す。

【００２９】ここで、タイムスタンプ５の値は書込み処理が完了した段階でインクリメントされる。このように、多数の細かいディスク書込み処理を１回で一括して実行できるため、ディスク書込み性能を向上させることができる。

（データ更新処理時の詰替え処理）次に、図７を参照してデータ更新処理時の詰替え処理について説明する。

【００３０】データ更新処理時では、旧データの領域を直接書き換えるのではなく、更新データを蓄積し、ディスク装置２内に予め用意した別の空領域に一括して書込む方法では空領域が常に存在することが必須である。このため、ホストシステム１０からのディスクアクセスが空いている間に、既に他の領域にデータが書込まれて無効になっているデータを寄せ集めて空領域を作る必要がある。この処理を詰替え処理と呼ぶ。この詰替え処理は、無効ブロック判定処理とストライプ統合の２つの処理からなる。

【００３１】無効ブロック判定処理の具体例として、図７に示すように、ホストシステム１０からのデータ書込み要求において、１ブロックサイズの書込み順序を想定する。図７において、Ｌ１８などの「Ｌ××」はホストシステム１０から渡される論理ブロック（論理アドレス）を意味し、Ｓ１などの「Ｓ××」は書込み順番を表わす。本実施形態では、書込みバッファ６は１５論理ブロックのデータを保持できる。最初のＳ１～Ｓ１５の書込みデータが、１つのストライプ（ＳＴ１）にまとめられて、タイムスタンプＴ１が付加されてディスク装置２の空領域に書き出される。同様に、Ｓ１６～Ｓ３０の書込みデータが別のストライプ（ＳＴ２）としてタイムスタンプＴ２が付加されて別の空領域に書き出される。なお、書き出し毎に、タイムスタンプ５はインクリメントされるので、「Ｔ１＜Ｔ２」の関係がある。

【００３２】ここで、図７から明らかなように、論理ブロックＬ９、Ｌ１８のデータはタイムスタンプＴ１のストライプではＳ５、Ｓ２として、タイムスタンプＴ２のストライプではＳ１９、Ｓ２１のブロックとして重複して存在する。書込まれた順番を考えると、Ｓ１９、Ｓ２１のデータブロックが有効であり、Ｓ５、Ｓ２のデータは無効と判定されなければならない。しかし、ここでは便宜上、使用した書込み順番Ｓ××は実際のディスク上には記録されていない。

【００３３】そこで、ストライプ内の論理アドレス・タグを使用して、この無効ブロック判定処理を行なう。図７の具体例において、２つのストライプＳＴ１、ＳＴ２の論理アドレス・タグＴＧ１、ＴＧ２の内容は、図８に示す通りである。図８から明らかなように、２つの論理アドレス・タグＴＧ１、ＴＧ２には同じ論理ブロックＬ９、Ｌ１８のデータが含まれている。ストライプＳＴ１のブロックＢ５、Ｂ２及びストライプＳＴ２のブロックＢ４、Ｂ６のいずれかのデータが無効である。また、論理アドレス・タグＴＧ２には有効なデータである論理ブロックＬ１８とＬ１９が存在している。さらに、論理アドレス・タグＴＧ１のタイムスタンプＴ１と論理アドレス・タグＴＧ２のタイムスタンプＴ２とを比較すると、「Ｔ１＜Ｔ２」の関係にあることから、ストライプＳＴ１のブロックＢ５、Ｂ２が無効であることが判定できる。以上のように、ディスク装置２内の論理アドレス・タグを参照することにより、無効なデータブロックを見つけることができる。なお、このような処理に必要な読み書きに伴うバッファは書込みバッファ７とは独立にした方が性能が向上する。当然ながら、書込みバッファ７と共有してもよい。

【００３４】ストライプ統合処理の具体例として、図９に示すように、２つのストライプＳＴ３、ＳＴ４を１つのストライプＳＴ５に統合する場合を想定する。図９に示すように、ストライプＳＴ３ではＢ２、Ｂ７、Ｂ８、Ｂ１２、Ｂ１３の５ブロックが有効であり、他の１０ブロックは無効（ハッチング）であるとする。同様に、ストライプＳＴ４ではブロックＢ１８、Ｂ１９、Ｂ２０、Ｂ２１、Ｂ２２、Ｂ２４、Ｂ２５、Ｂ２７、Ｂ２９の９ブロックが有効であり、他の６ブロックが無効（ハッチング）であるとする。２つのストライプの有効ブロックは合わせて１４ブロックしかないので、この２つのブロックを１つに統合することにより、結果として１つの空領域が作れる。

【００３５】ストライプ統合では図９に示すように、２つのストライプＳＴ３、ＳＴ４を揮発性メモリ３内に読みだし、有効なブロックだけを詰めて書込みバッファ６に移す。この場合に、ストライプＳＴ１とストライプＳＴ２の中にある有効なブロックの中から論理アドレスが連続になるようにする。こうする事により、統合されたストライプＳＴ５のブロックは、論理アドレスとしてだけでなく物理アドレスとしても連続したデータを保持できることになる。これに合わせて、図１０に示すように、論理アドレス・タグもＴＧ３、ＴＧ４から有効ブロックの論理アドレスだけを対応する位置に移し、新しい論理アドレス・タグＴＧ５を作成し、その時点のタイムスタンプ５を更新する。図１０から明らかなように、新しく作成された論理アドレス・タグＴＧ５に含まれる論理ブロック（アドレス）Ｌ１３、Ｌ１４とＬ２２とＬ２３は、ストライプＳＴ５上でも連続な領域となっている。すなわち、論理的にも物理的にも連続な領域が確保されていることになる。

【００３６】この具体例のように、１４個の有効ブロックを想定する場合には、さらにホストシステム１０から１つの書き込みブロックを有するストライプを完成させて、ディスク装置２の空領域に一括して書き込むことになる。このとき、論理アドレスの並替えを行なっても問題はない。この場合、ディスク領域は有効に活用されるが、ホストシステム１０からバーストでディスクアクセスがある場合には、書き込み処理を待たすために、ディスクアクセスが集中する可能性が高い。そこで、最後のデータブロックは空状態のままにして、アクセスが空いている間に書き込むことも可能である。このとき、論理アドレス・タグＴＧ５の最後のデータブロックに対する論理アドレスには－１等ＮＵＬＬアドレスを入れることにより、データが入っていないことを表し、データが格

納されているデータブロックに対して論理アドレスの並び変えをしても問題は発生しない。

【００３７】次に、書込まれたデータブロックの読出し動作について説明する。

【００３８】前述の詰替え処理における無効ブロック判定処理を、ディスク装置２内の全ストライプの論理アドレス・タグに対して行うことにより、全論理アドレスに対する有効ブロックの物理的位置を検出できる。従って、原理的には、ホストシステム１０から読出しブロックの論理アドレスを受け取る度に、全ストライプのチェックを行うことにより、読出すべき物理ブロックを見つけ出すことが出来る。しかし、この方法ではブロック読出し処理に膨大な時間を要するため実用的ではない。

【００３９】そこで、システム起動時にだけ、全ストライプの論理アドレス・タグの調査を実行して、揮発性メモリ３上に論理アドレスから物理アドレスへの変換マップ８を作成する。ホストシステム１０からの読出し要求に対しては、当該変換マップ８を使用して有効ブロックへのアクセスを実行する。これにより、常に論理アドレス・タグの調査をしなくても良く、読出し時に性能が大きく低下することはない。また、この変換マップ８は何時でも全ストライプを調査することで再生できるため、従来のように電源障害に備えて不揮発メモリ４上に格納する必要はない。

（変換マップの構成）以下図１１から図１３を参照して、本実施形態の変換マップ８の構成について説明する。

【００４０】変換マップ８は、図１１に示すように、各論理アドレスに対するブロックが格納されているストライプ番号ＳＴ＃、当該ストライプ内のブロック番号ＢＬＫ＃、さらにタイムスタンプＴＳ＃をテーブル形式で保持しているテーブル情報である。制御装置１は、論理アドレスＬ０～Ｌｎにより変換マップ８を参照して、ＳＴ＃とＢＬＫ＃から実際の物理アドレスを求める。

【００４１】ここで、システム起動時には、全ストライプの論理アドレス・タグを調査し、ディスク装置２内の全てのストライプに対して、無効ブロックを検出する。次に、全てのストライプに対してストライプ統合処理を実行する。この場合には、各論理アドレスに対応するブロックが格納されているストライプ番号ＳＴ＃と、当該ストライプ内のブロック番号ＢＬＫ＃と、タイムスタンプＴＳ＃とから、論理アドレスの連続性を考慮して、ストライプ番号ＳＴ＃、ブロック番号ＢＬＫ＃をできる限り連続的になるように再配置する。これにより、連続したデータの読込み時に、ストライプ番号ＳＴ＃が異なったり、同じストライプ番号ＳＴ＃でブロック番号ＢＬＫ＃が不連続になるという可能性が少なくなり、ブロック毎に行なっていた読込み要求は、書込みバッファ６とバッファ管理テーブル７との関係からも分かるように、複数ブロックを一度に読込むことが可能になり、性能低下を軽減できる。

【００４２】図１１に示す変換マップの具体例では、ディスク装置２へ書込み処理または更新処理のためにアクセスする場合には、変換マップ８の全てのエントリを参照することになる。この方法では、全てのエントリを検索しなければならず、検索中に変換マップ８の全体を排他制御しなければならないといった制約があるため、効率的なアドレス変換であるとはいえない。そこで、効率的なアドレス変換を可能にするために、論理アドレスをハッシュキーにした変換マップ（図１２を参照）や論理アドレスを２分岐検索する変換マップ（図１３を参照）を作成することが望ましい。これらの変換マップを使用することにより、効率的なアドレス変換が可能となる。

【００４３】論理アドレスをハッシュキーにした変換マップの場合には、頻繁に使用される論理アドレスのエントリをハッシュリストの先頭に配置するように作成する。このことにより、読み出したデータを更新するときに、更新性能が向上する。本構造で変換マップを作成する場合には、検索時間を考慮して、根の論理アドレスに考慮する必要がある。図１３の具体例では、コンピュータシステムで取扱い可能な論理アドレスの中間値を使用した２分岐による本構造であるが、多分岐による本構造も可能である。

【００４４】また、システム起動時の変換マップの作成は、調査した論理アドレス・タグの全論理アドレスについて、テーブルのタイムスタンプ５より論理アドレス・タグのタイムスタンプ５が大きいときだけ、そのストライプ番号ＳＴ＃と対応するブロック番号ＢＬＫ＃を各々の変換マップに応じてテーブル、ハッシュリスト、本構造に登録する。この調査を全ストライプについて行なえば、有効ブロックだけを指す変換マップが作成される。この場合にも、ストライプ統合処理と同様の処理をすることにより、連続したデータの読込み処理時に、ストライプ番号ＳＴ＃が異なったり、同じストライプ番号ＳＴ＃でブロック番号ＢＬＫ＃が不連続となる可能性が少なくなり、性能を低下することがなくなる。更に、ディスク装置２にストライプを書込む毎に、その論理アドレス・タグに対して同様の処理を行なうことにより、この変換マップは常に有効なブロックを指す。また、ディスクアクセスが空いているときに、各ストライプの論理アドレス・タグと変換マップとを比較検査することにより、離散しているデータを同一のストライプや連続したブロックに統合することができる。更に、メモリ障害等でこの変換マップが不正な値になっても検出可能である。

【００４５】以上のような変換マップ８は、ディスク装置２の記憶容量に応じて、書き込み処理の時間的順序を維持するために必要なタイムスタンプ５と、論理アドレスから物理アドレスへのアドレス変換を行なうため揮発性メモリ３の記憶容量とが必要となる。このため、記憶容量が大きなディスク装置２の場合には、メインメモリ

上の揮発性メモリ３が多く必要となるため実際的ではない。そこで、最新データの位置情報を管理する間接マップの一部を、ディスク装置内の空領域に格納し、変換するデータとその論理アドレスとが間接マップに存在しない場合には、保存してあるディスク装置２内の空領域から間接マップを読みだし、間接マップを再作成することが有効になる。

【００４６】ここで、間接マップを再作成する場合について説明する。論理アドレスから物理アドレスに変換する度に、ディスク装置に保存してある間接マップを読出して再作成する方式では、コンピュータシステムの性能上から容認しがたいものになってくる。そこで、ディスク装置内に保存してある間接マップの一部に対して、高頻度の読出し要求が発生しないようにすることが重要となる。そのためには、各論理アドレスに対応する間接マップのエントリのタイムスタンプの古いものから順番にディスク装置２へ保存することにより、最適な間接マップを構成することが可能である。

【００４７】以上のように、変換マップ８の作成処理における要素は論理アドレス・タグの検査である。故に、大容量のディスク装置２のように論理アドレス・タグ数が多い場合、電源障害やシステム起動時の変換マップ作成に長時間を要する。特に、図２に示すように、論理アドレス・タグ・ブロックが１台のディスクドライブ２４に集中すると、システム起動時には当該ディスクドライブ２４にアクセスが集中し、論理アドレス・タグの調査を並列に行うことが不可能となる。そこで、図１４に示すように、ストライプにより論理アドレス・タグが格納されるディスクドライブを４台に分散して並列に論理アドレスタグを調査することにより、この変換マップ作成に要する時間を１／４に短縮化できる。

（セグメント分割管理）図１５と図１６はディスク装置２の記憶領域を複数のセグメントに分割管理するセグメント分割管理方式を示す図である。この方式により、変換マップ８の作成に必要な論理アドレス・タグの検査数を削減することができる。図１６にセグメント分割方式におけるディスク装置の記憶領域の構成を示す。図１６に示すように、ディスク装置の記憶領域は、ストライプを単位としてセグメント管理情報（ハッチング）と４つのセグメントに分割される。ここで、セグメントとは書込みバッファデータの一括書込み処理や詰め替え処理のディスク書込みがある期間集中して行われる単位領域のことである。例えば、セグメント２がディスク書込みの対象である間は、セグメント１、３、４には書込み処理が実行されないように空領域の選択を制御する。

【００４８】また、あるセグメントの空領域が少なくなりディスク書込みを他のセグメントへ切替えるときにはセグメント管理情報をディスク装置に保存する。セグメント管理情報は図１５に示すように、セグメント番号と切替え時変換マップから構成される。セグメント番号とは切替え先のセグメント番号で、切替え時変換マップとはセグメントを切替える時点での揮発性メモリ３上の変換マップ８の状態を示す。

【００４９】なお、切替え時変換マップはセグメントが切替える度に全て上書きするのではなく、直前のセグメントに書込まれた論理アドレスのエントリだけを書き戻せばよい。従って、前回のセグメント切替え時にタイムスタンプを覚えておき、変換マップのタイムスタンプを比較することにより、直前のセグメントに書込まれた論理アドレスを判定できる。

【００５０】このセグメント分割方式では、セグメント切替え時にセグメント管理情報を保存している。よって、セグメント切替え時の変換マップをセグメント管理情報から読み出して、その後でセグメント管理情報のセグメント番号で指示されるセグメントの論理アドレスタグだけを検査するだけで、全論理アドレス・タグを検査した場合と同じ変換マップが再現できる。従って、この方式により必要な論理アドレス・タグの検査数は１セグメント分で良く、この例では変換マップの作成に要する時間を１／４に短縮化することが可能となる。

【００５１】更に、不揮発性メモリ４上にセグメント内の全ストライプに対応したビットマップを用意して、セグメント切替え時にはこのビットマップをクリアし、一括書込みや詰替えの書込み時には書込んだストライプに対応するビットを“１”にセットする。これによりセグメントを切替えてから変更のあるストライプだけがビットマップが“１”になる。従って、変換マップ作成時にこのビットマップを参照し、変更の有ったストライプの論理アドレスタグだけを検査することで検査数をさらに減らせ、変換マップ作成に要する時間を更に短縮化できる。

【００５２】通常、論理アドレス・タグのサイズは５１２～１０２４バイトであり、ディスクのシーケンシャルアクセスとランダムアクセスに約５０倍の性能差がある。図２に示す方式では論理アドレス・タグの情報が各ストライプ毎にとびとびに存在するので、変換マップの作成では時間のかかるランダムアクセスを実行している。そこで、図１７に示すように、論理アドレス・タグだけを連続して格納する専用タグ領域を（セグメント分割する場合は、各セグメント毎に）用意し、５０倍も高速なシーケンシャルアクセスで論理アドレスタグを読み出せるようにする。そして、ホストシステム１０からのデータの一括書込み処理や詰替え処理時には、空領域だけでなく対応する専用タグ領域にも論理アドレス・タグを書込むようにする。この方式であれば、１ストライプ当たり４回のディスク書込み処理に対して、専用タグ領域への論理アドレス・タグの書込み処理のために、書込み操作が１回増えることになる。しかし、変換マップ作成が５０倍も高速になるので、ディスク装置の立ち上がり時間が問題となるときには非常に有効な手段である。

専用タグ領域への書込み時間を最小にするためには、専用タグ領域は図17に示すように、対象領域を中心に設定することにより、ディスクドライブのシーク時間を短縮化することが望ましい。また、通常ではディスク装置2はセクタ（512バイトなど）単位の書込み処理を実行するため、専用タグ領域内の論理アドレスタグはセクタ単位で割り付けて、論理アドレス・タグの書込み処理時には読出し処理を不要にすることが望ましい。

（タイムスタンプの説明）図1に示すように、タイムスタンプ5は揮発メモリ3上に記憶されている。このため、システムの電源障害などにより揮発メモリ3上のタイムスタンプ5が消去する可能性がある。そこで、変換マップ8の場合と同様にして、システム起動時にだけ全ストライプの論理アドレス・タグを調査し、一番大きなタイムスタンプ5の次の値を揮発メモリ3上のタイムスタンプ5にセットする。なお、変換マップの作成処理の説明で述べた時間短縮手法がそのままタイムスタンプの再生にも適用できる。

【0053】また、タイムスタンプ5はディスク装置2に書込む度に、インクリメントされて、ディスク上の書込み順序の判定処理のみに使用される。具体例として、タイムスタンプ5が24ビットのカウンタで構成される場合を想定する。24ビットカウンタでは、16M回の書込み処理でカウンタが一周してクリアされてしまう。そこで、一般的には有効なタイムスタンプ5の最小値を基準として、それより小さい値は16Mを加えて比較して判定する。この最小値も同様にシステム起動時にだけ全ストライプの論理アドレス・タグを調査して求める。しかし、この手法が使えるのはタイムスタンプの最大値が最小値を追い越さないこと、つまり、タイムスタンプの最大値と最小値との差が24ビットで表わせる範囲以内であることを前提にしている。従って、タイムスタンプ5が一周前に必ず全ストライプを更新してタイムスタンプ値を新しく更新する必要がある。これには、無効ブロックが少なくても予め設定した書込み回数の間に更新されなかったストライプを詰替えの対象として選ぶように制御するか、無効ブロックの論理アドレスをNULLアドレスにした、そのストライプの論理アドレスタグだけを書き換える。NULLアドレスを使う方法は論理アドレス・タグ・ブロックの書き換えであるので詰替え処理と比較して常に軽い処理である。

【0054】尚、前記の具体例では、無効ブロック判定処理に2つストライプST1、ST2の論理アドレス・タグを相互に比較して判定する方法についてのみ説明したが、全無効ブロックを調べるには2つのストライプ間の全組み合わせを調べなければならない。しかし、変換マップがあれば論理アドレス・タグ内の各論理アドレスについて、有効データを指示している変換マップのタイムスタンプとそのストライプのタイムスタンプとを比較し、値が小さいブロックを無効ブロックと判定することができる。

（本実施形態の変形例）図18は本実施形態の変形例を示す図である。本実施形態は、RAID0方式を適用したシステムを想定している。これに対して、本変形例はパリティデータを使用する冗長性ディスク構成のRAID5方式を適用したシステムである。

【0055】本システムは、図18に示すように、図1のシステムに対して、冗長用のディスクドライブ25が追加された構成である。なお、これ以外の制御装置1、ディスク装置2（ディスクドライブ21、22、23、24）、揮発性メモリ3、不揮発性メモリ4、タイムスタンプ5、書込みバッファ6、およびバッファ管理情報（テーブル）7は、図1に示す本実施形態と同じ機能を有する要素である。

【0056】以下、本変形例のシステムの動作を、本実施形態との差異を中心にして説明する。書込み処理では、ホストシステム10からの書込みデータが1ストライプ分に1ブロック少ない数（N＊K－1）だけ書込みバッファ6に蓄積された段階で、制御装置1はそれらのデータをディスクドライブ21～25に書込み処理していく。このとき、最後の書込みブロックとして、バッファ管理管理テーブル7に格納された各ブロックの論理アドレスと揮発性メモリ3上のタイムスタンプ5とから論理アドレス・タグ・ブロックを作成するまでは処理は、本実施形態の場合と同様である。

【0057】この後、制御装置1は、図19に示すように、論理アドレス・タグ・ブロックを加えた1ストライプ分のデータからストライプユニット毎の排他論理輪演算（XOR）を実行し、パリティデータ（P）のストライプユニットを作成する。そして、このパリティデータ付きのストライプのデータを一括して、ディスクドライブ21～25の空領域に同時に書込む。本実施形態のように、可変長のブロック単位での読込み処理を行なう場合には、書込むための空領域は、各ディスクの現在のストライプ位置からの差分が最も少なくなる空領域に近いストライプを選択する。このような選択方法により、各ディスクのシーク時間による書込み処理の応答時間を均等にすることができる。ストライプ単位での読込みによる場合には、このようなことにはならないので問題がない。

【0058】また、タイムスタンプ5の値は書込み処理が完了した段階でインクリメントされる。このように多数の細かいディスク書込み処理を1回にまとめられるのに加え、パリティ計算に必要な旧データや旧パリティデータのブロックを読込む必要がないので、さらにディスクアクセス回数を減少することができる。なお、ストライプ詰替え処理時のディスク書込み処理でも、同様にしてパリティデータ付きのストライプを作成してからディスク装置2に書込む（図19を参照）。

【0059】冗長性ディスク構成（パリティ）のRAI

Ｄ５方式のシステムでは、１台のディスクドライブが故障しても、故障したディスクドライブのデータはストライプを構成する他ディスクドライブのデータとパリティデータとのＸＯＲ演算を実行することにより再現することができる。しかし、システム起動時に一台故障していた場合には、論理アドレス・タグを格納していないディスクドライブのデータも読出して論理アドレス・タグを再生してから検査するため、変換マップ８の作成処理にかなり時間を要し、システム起動が完了するまでの時間が大幅に増大してしまう。

【００６０】そこで、図２０に示すように、ストライプを構成するデータブロックを１つ減らして２つのディスクドライブに同じ論理アドレス・タグを書込むように制御する。これにより、ディスクドライブの１台が故障しても、変換マップ８の作成時には残っている方の論理アドレス・タグを読出すことができるので、システム起動に要する時間の大幅な増大を回避できる。

【００６１】また、変換マップ作成の高速化のために専用タグ領域を活用する場合、図２１に示すように、論理アドレス・タグが専用タグ領域に格納されるディスクドライブと、ストライプに格納されるディスクドライブとが異なるように、専用タグ領域の論理アドレス・タグの割り付け処理を制御することにより、ストライプ内の論理アドレス・タグは１つでよい。

【００６２】なお、専用タグ領域へ論理アドレス・タグを書込む場合も、パリティデータによるディスク障害対策を行うと、従来１回の書込み操作の増大で済んでいたものが、２回の書込み処理と２回の読出し処理が必要になって、一括書込み処理時やストライプ詰替え処理時のディスク書込み動作のオーバヘッドが大きく増大する。従って、この専用タグ領域の情報はパリティデータを使用した障害対策を実行しない方が望ましい。専用タグ領域の情報は変換マップ高速化のためであり、故障したディスクドライブの専用タグ領域に格納されていた論理アドレス・タグは、ストライプ中のものを（ランダムアクセスで）見れば良いので問題はない。また、ランダムアクセスで検査する論理アドレス・タグは１／５だけであるので、変換マップ作成の高速化の効果はある。

【００６３】なお、本実施形態は、ＲＡＩＤ方式のディスク記憶システムに適用した場合について説明したが、これに限る事なく、光磁気ディスク等を使用したシステムにも適用することができる。また、シーケンシャル書込み処理とランダム書込み処理とで大きく性能が異なるディスク記憶システムや、小ブロックの更新処理では２回の読出し処理と２回の書込み処理とが必要なパリティによる冗長性を持たせたＲＡＩＤ構成のディスク記憶システムにも適用できる。

【００６４】

【発明の効果】以上詳述したように本発明によれば、ディスク装置から読出すべきデータを不揮発性メモリ上の書込みバッファの空領域に物理的に連続したブロック単位に分割して詰めて格納する方式であるため、一括したアドレスに変換できるため、読出しのための入出力要求が分割される必要がない。また、ディスク装置の容量に依存しないでメインメモリ上に変換マップを作成することができるため、メインメモリの記憶領域を効率的に使用することができる。本発明を特に高速の書込み方法を採用したＲＡＩＤ方式のディスク記憶システムに適用すれば、特にデータ更新処理に関する性能の向上を図ることが可能となる。

【図面の簡単な説明】

【図１】本発明の実施形態のＲＡＩＤ方式のディスク記憶システムの概念的構成を示すブロック図。

【図２】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図３】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図４】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図５】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図６】本実施形態の書込み動作におけるディスク装置の空領域の格納内容を説明するための図。

【図７】本実施形態の書込み動作における詰替え処理を説明するための図。

【図８】図７の具体例におけるストライプＳＴ１、ＳＴ２の論理アドレス・タグＴＧ１／ＴＧ２の内容を示す図。

【図９】図７の具体例におけるストライプ統合処理を説明するための図。

【図１０】図９のストライプ統合処理において、論理アドレスタグＴＧ３／ＴＧ４から論理アドレスタグＴＧ５を作成する場合の具体例を示す図。

【図１１】本実施形態の変換マップの構成を説明するための図。

【図１２】本実施形態の変換マップの構成を説明するための図。

【図１３】本実施形態の変換マップの構成を説明するための図。

【図１４】本実施形態に関係する論理アドレス・タグを分散配置した具体例を説明するための図。

【図１５】本実施形態に関係するセグメント分割管理を説明するための図。

【図１６】本実施形態に関係するセグメント分割管理を説明するための図。

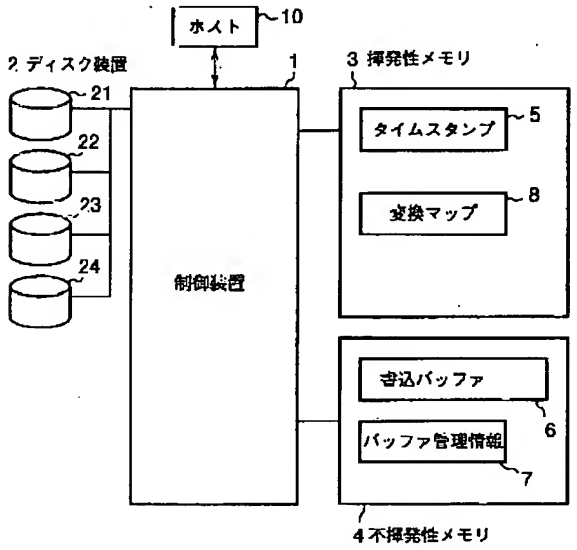
【図１７】本実施形態に関係する専用タグ領域の内容を示す図。

【図１８】本実施形態の変形例に関係するＲＡＩＤ５方式のディスク記憶システムの概念的構成を示すブロック図。

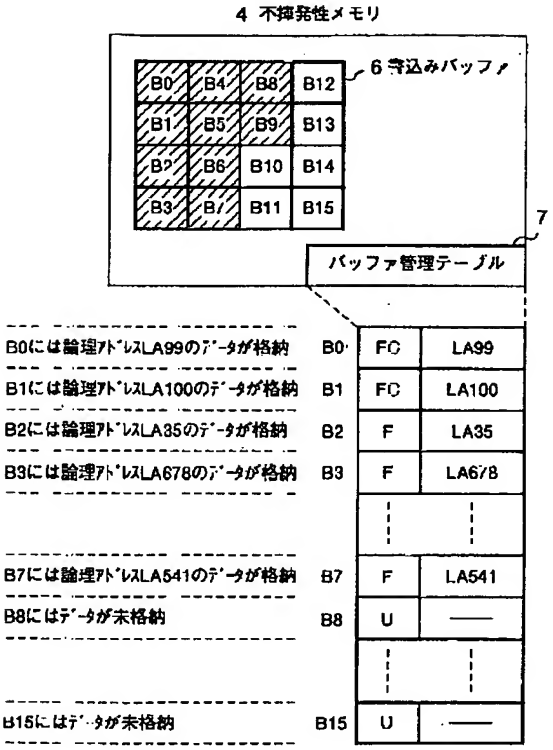
【図１９】本変形例の動作を説明するための概念図。
【図２０】本変形例に関係する論理アドレス・タグの作成方法を示す図。
【図２１】本変形例に関係する専用タグ領域を割付け方法を説明するための図。
【図２２】従来のディスク記憶システムのデータ更新方法を説明するための概念図。
【図２３】本発明の実施形態のＲＡＩＤ方式のディスク記憶システムの概念的構成を示すブロック図。
【符号の説明】

１…制御装置（ＣＰＵ）
２…ＲＡＩＤ方式のディスク装置
３…揮発性メモリ（メインメモリ）
４…不揮発性メモリ
５…タイムスタンプ
６…書込みバッファ
７…バッファ管理情報（テーブル）
１０…ホストシステム
２１～２５…ディスクドライブ

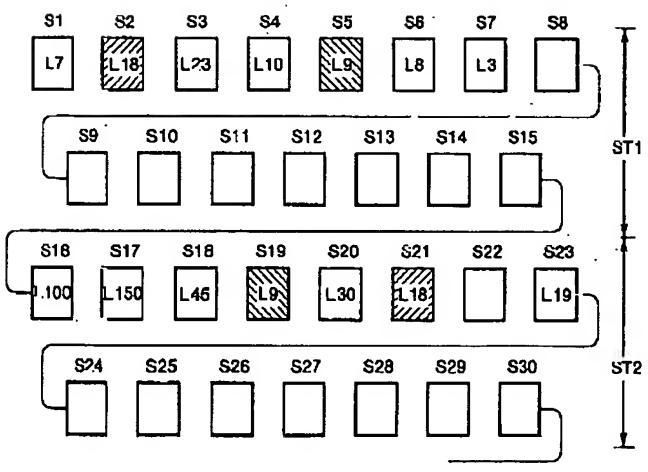
【図１】



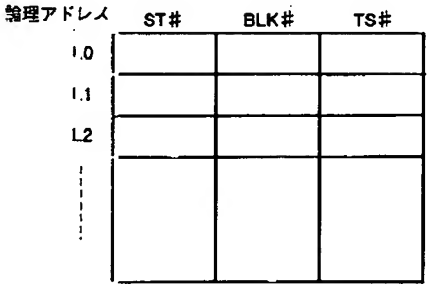
【図２】



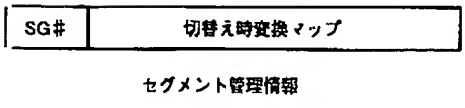
【図７】



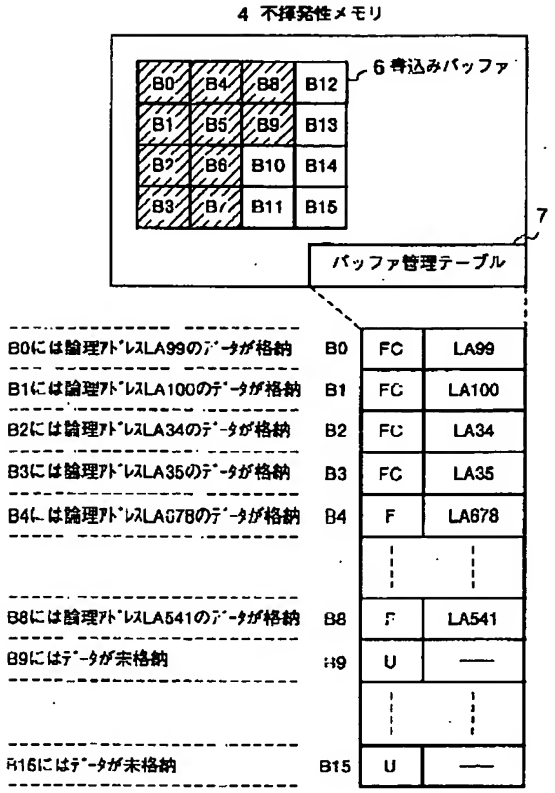
【図１１】



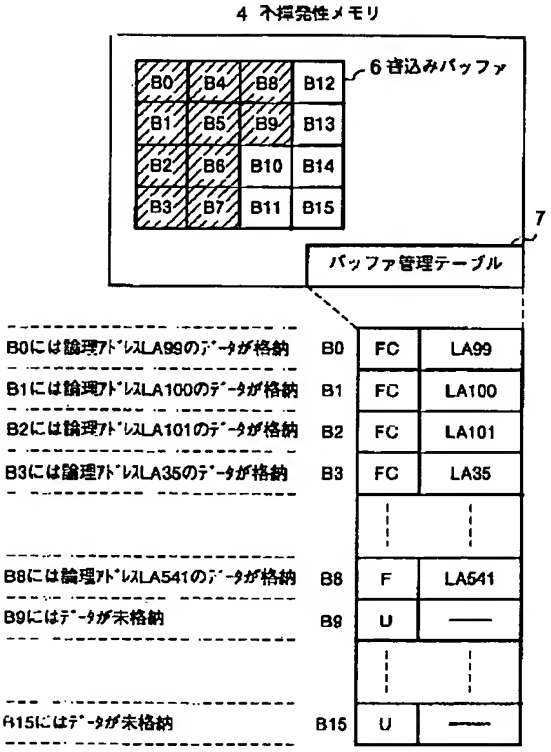
【図１５】



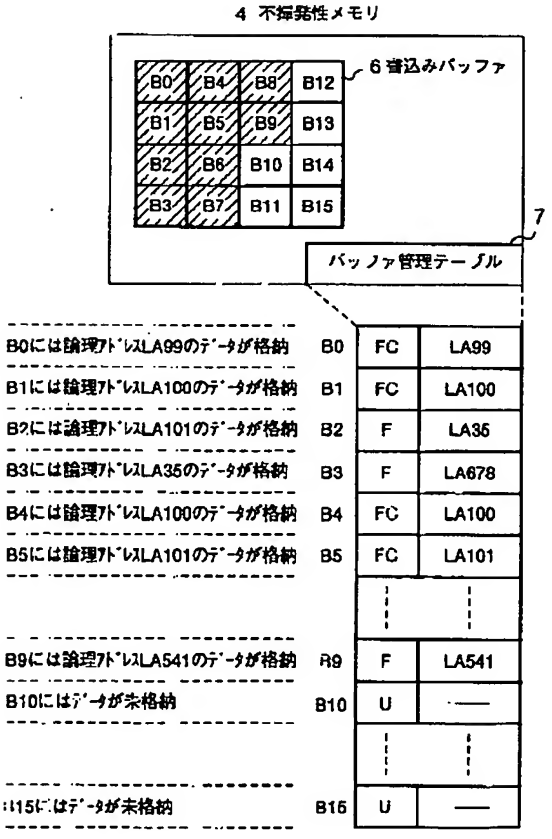
【図3】


4 不揮発性メモリ
6 書込みバッファ
B0 B4 B8 B12
B1 B5 B9 B13
B2 B6 B10 B14
B3 B7 B11 B15
7
バッファ管理テーブル
B0には論理アドレスLA99のデータが格納 B0 FC LA99
B1には論理アドレスLA100のデータが格納 B1 FC LA100
B2には論理アドレスLA34のデータが格納 B2 FC LA34
B3には論理アドレスLA35のデータが格納 B3 FC LA35
B4には論理アドレスLA678のデータが格納 B4 F LA678
B8には論理アドレスLA541のデータが格納 B8 F LA541
B9にはデータが未格納 B9 U ——
B15にはデータが未格納 B15 U ——

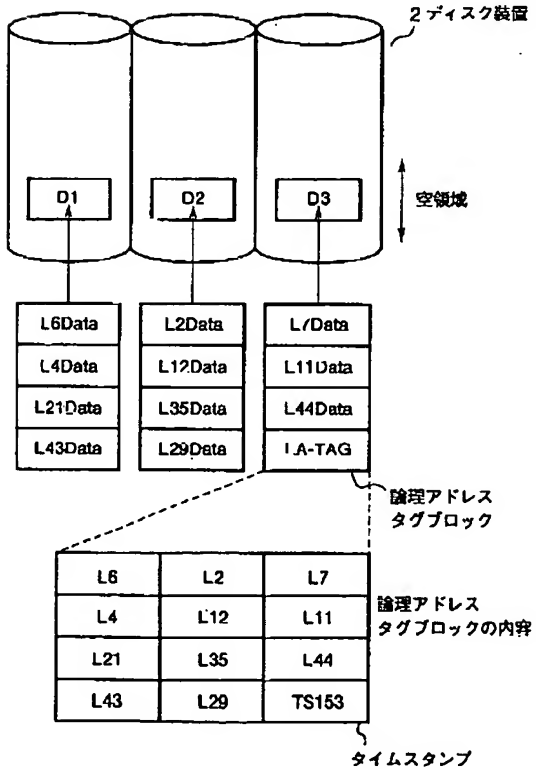
【図4】


4 不揮発性メモリ
6 書込みバッファ
B0 B4 B8 B12
B1 B5 B9 B13
B2 B6 B10 B14
B3 B7 B11 B15
7
バッファ管理テーブル
B0には論理アドレスLA99のデータが格納 B0 FC LA99
B1には論理アドレスLA100のデータが格納 B1 FC LA100
B2には論理アドレスLA101のデータが格納 B2 FC LA101
B3には論理アドレスLA35のデータが格納 B3 FC LA35
B8には論理アドレスLA541のデータが格納 B8 F LA541
B9にはデータが未格納 B9 U ——
B15にはデータが未格納 B15 U ——

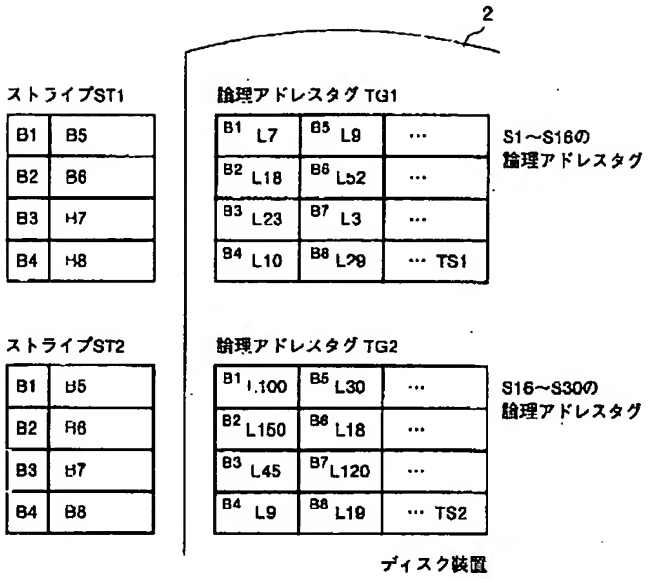
【図5】


4 不揮発性メモリ
6 書込みバッファ
B0 B4 B8 B12
B1 B5 B9 B13
B2 B6 B10 B14
B3 B7 B11 B15
7
バッファ管理テーブル
B0には論理アドレスLA99のデータが格納 B0 FC LA99
B1には論理アドレスLA100のデータが格納 B1 FC LA100
B2には論理アドレスLA101のデータが格納 B2 F LA35
B3には論理アドレスLA35のデータが格納 B3 F LA678
B4には論理アドレスLA100のデータが格納 B4 FC LA100
B5には論理アドレスLA101のデータが格納 B5 FC LA101
B9には論理アドレスLA541のデータが格納 B9 F LA541
B10にはデータが未格納 B10 U ——
B15にはデータが未格納 B15 U ——

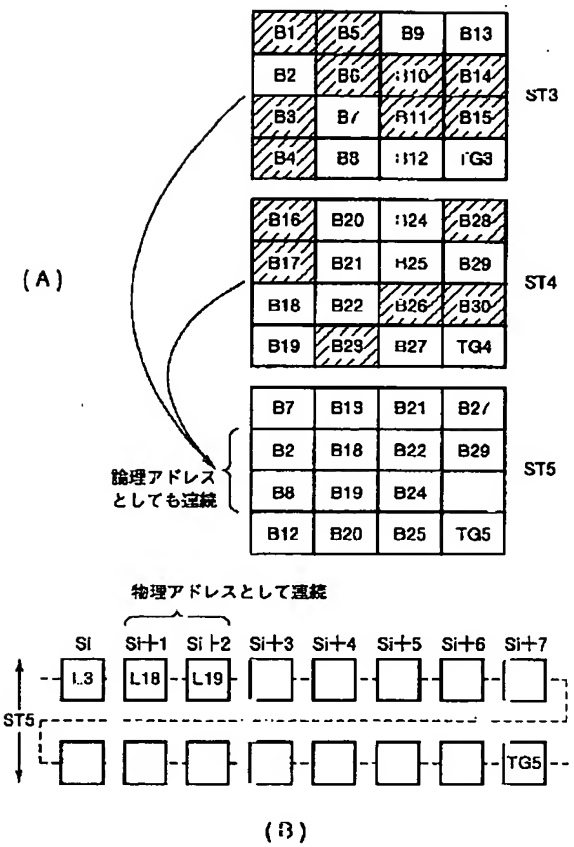
【図6】


2 ディスク装置
D1
D2
D3
空領域
L6Data L2Data L7Data
L4Data L12Data L11Data
L21Data L35Data L44Data
L43Data L29Data LA-TAG
論理アドレス
タグブロック
L6 L2 L7
L4 L12 L11
L21 L35 L44
L43 L29 TS153
論理アドレス
タグブロックの内容
タイムスタンプ

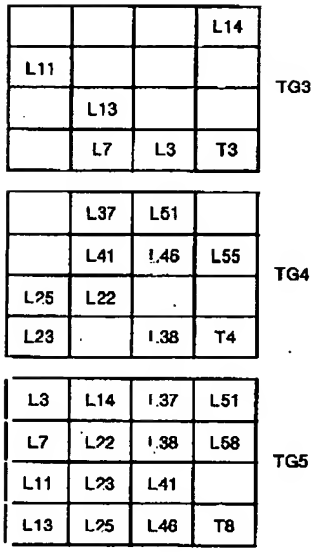
【図8】



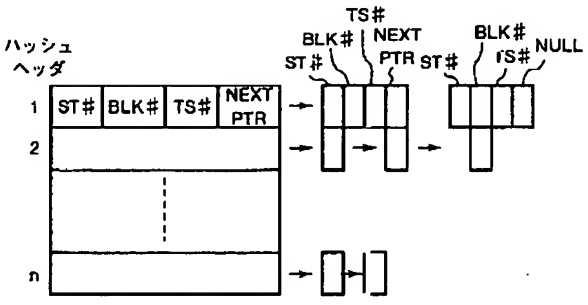
【図9】



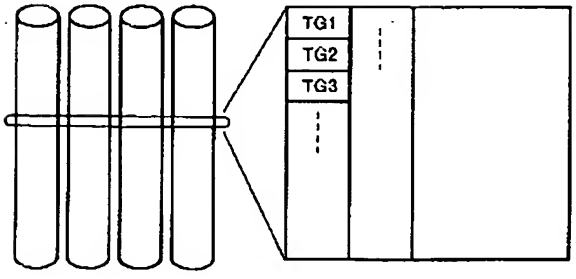
【図10】



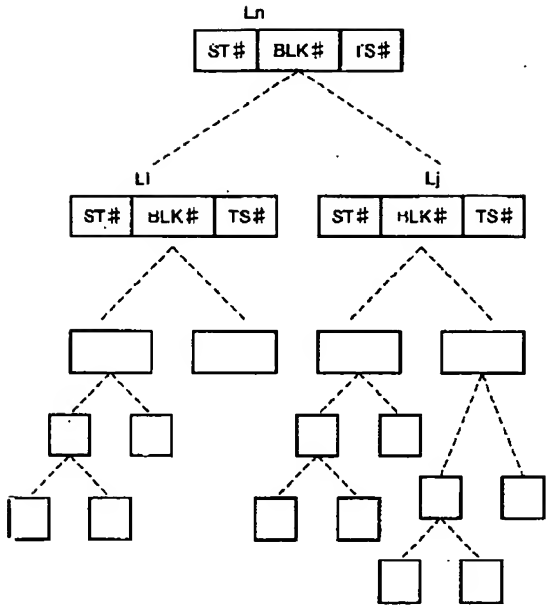
【図12】



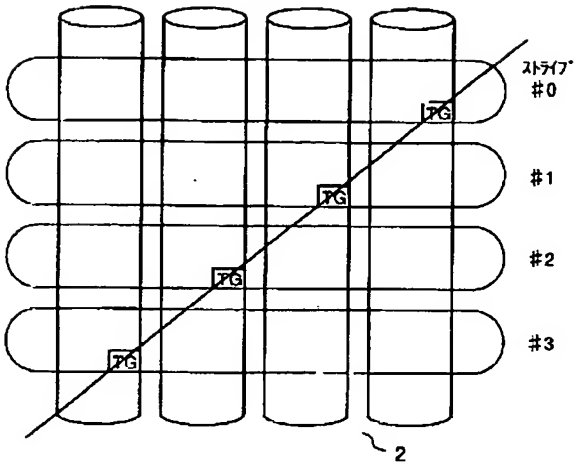
【図17】



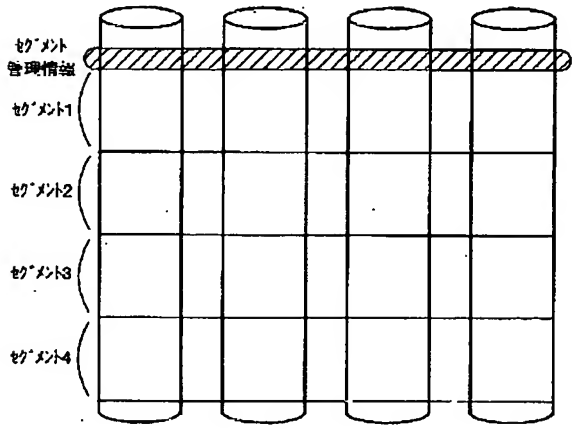
【図13】



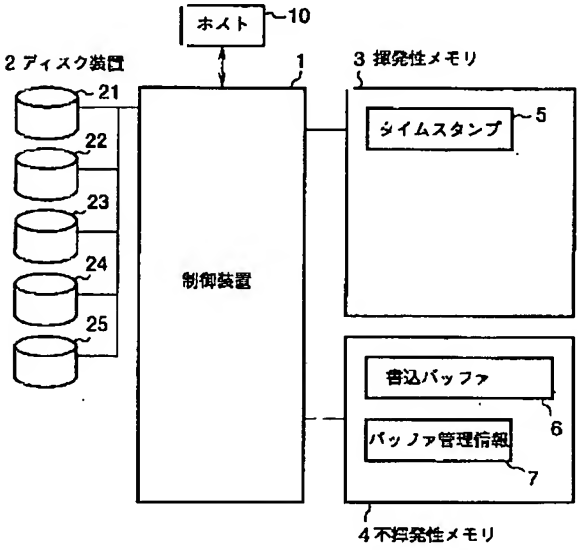
【図14】



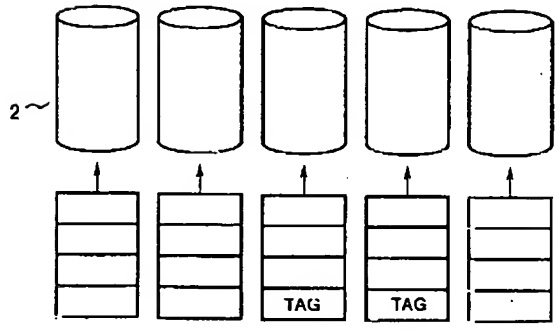
【図16】



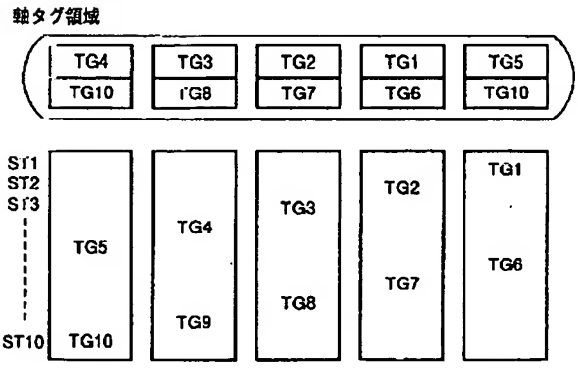
【図18】



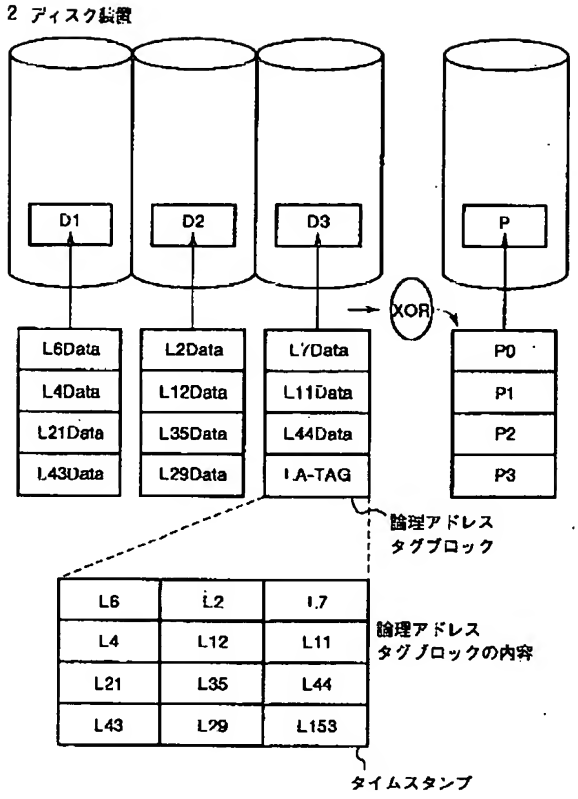
【図20】



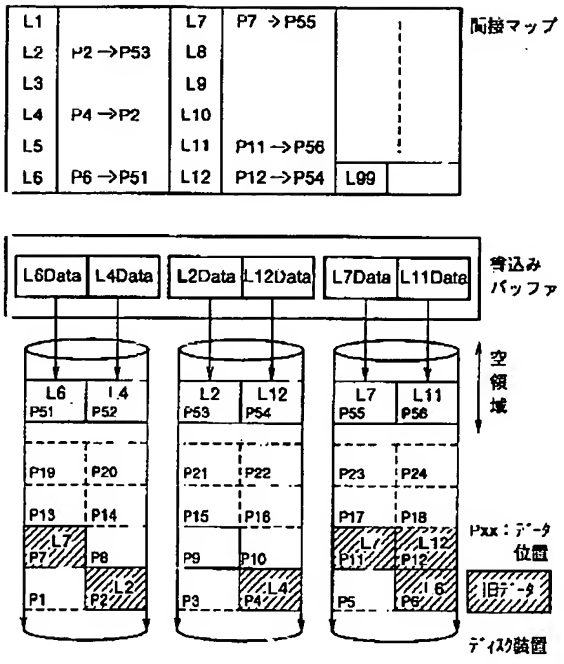
【図21】



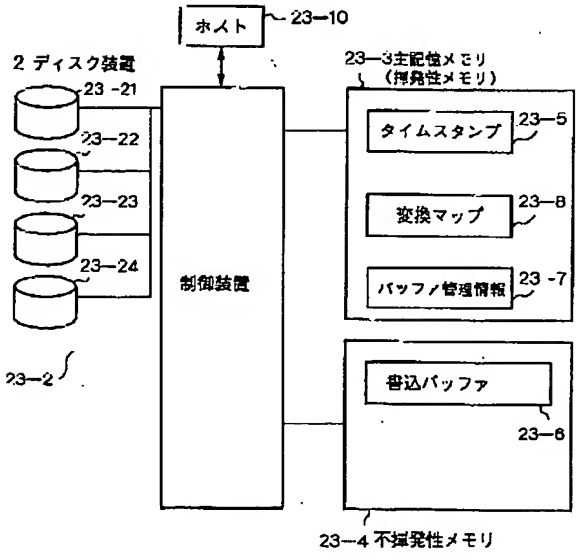
【図19】


2 ディスク装置
D1
D2
D3
P
XOR
L6Data
L4Data
L21Data
L43Data
L2Data
L12Data
L35Data
L29Data
L7Data
L11Data
L44Data
LA-TAG
P0
P1
P2
P3
論理アドレス
タグブロック
L6
L2
L7
L4
L12
L11
L21
L35
L44
L43
L29
L153
論理アドレス
タグブロックの内容
タイムスタンプ

【図22】


L1
L2
L3
L4
L5
L6
P2→P53
P4→P2
P6→P51
L7
L8
L9
L10
L11
L12
P7→P55
P11→P56
P12→P54
L99
間接マップ
L6Data
L4Data
L2Data
L12Data
L7Data
L11Data
書込み
バッファ
空
領
域
L6
P51
L4
P52
L2
P53
L12
P54
L7
P55
L11
P56
P19
P20
P13
P14
P21
P22
P15
P16
P23
P24
P17
P18
L7
P7
P8
L2
P2
P9
P10
L4
P4
L7
P11
L12
P12
L6
P6
P1
P3
P5
Pxx：データ
位置
旧データ
ディスク装置

【図23】


ホスト
23—10
2 ディスク装置
23—21
23—22
23—23
23—24
23—2
制御装置
23—3主記憶メモリ
（揮発性メモリ）
タイムスタンプ
23—5
変換マップ
23—8
バッファ管理情報
23—7
書込バッファ
23—6
23—4 不揮発性メモリ